To appear in Proc. IEEE Autotest Conference 1997.

QSI's Integrated Diagnostics Toolset

Dr. Somnath Deb, Dr. Krishna R. Pattipati and Roshan Shrestha

Qualtech Systems Inc., 66 Davis Road, Storrs, CT 06268

Phone/Fax: (860) 423-2099/8422. Email: deb@teamqsi.com

Abstract: The QSI integrated tool set, consisting of TEAMS, TEAMS-RT, TEAMATE and HARVESTER, offers a comprehensive solution to integrated diagnosis of systems with many components (modules, boards, replaceable units, etc.) that are subject to failure. The software tool set automates the DFT, FMECA, on-line monitoring, off-line diagnosis, and maintenance data management tasks. Integration is achieved via a common model-based approach wherein a consistent model is used across different tools at various stages of a system's life-cycle. In this paper, we present an overview of the Integrated Toolset, with examples of its real-world applications in model-based TPS development, real-time process monitoring, and PIMA.

1. INTRODUCTION

The QSI Testability Engineering tool, TEAMS, was introduced in 1992 and is being used in the V22, F22, Comanche, JSF and other projects primarily for DFT. As our customer base moves towards deployment of these aircraft, we are building the methodology and the tools that will support these systems in the field. Current research at Qualtech Systems, Inc. (QSI) is focused on developing a complete solution package for Integrated Diagnostics (ID) that addresses all the facets of Design for Testability (DFT), Test Program Set (TPS) development and field maintenance. The solution involves four key innovations.

First, since the design process involves multiple disciplines, a simple, yet efficient, knowledge representation (i.e., system models that are easy to create, verify and validate) is essential. In [1], we introduced the multisignal modeling methodology, which has the benefit of capturing the necessary information about the system in a simple, intuitive format. The information captured in multi-signal models (structure, reliability data, and basic functional information or signals) are basic attributes of a system that can be readily extracted from design data and product specifications. Hence, building multi-signal models is not expensive; yet these models capture sufficient diagnostic information for DFT, Failure Modes, Effects and Criticality Analysis (FMECA), TPS development, on-line monitoring and Portable Intelligent Maintenance Aid (PIMA). Additional information can be easily added as design becomes more refined and detailed, and models can be updated when designs are revised. This flexibility and versatility ensure that the models can evolve with the product, from concept→ schematics→ fielded product.

Second, an integrated tool set based on multisignal models that automates the DFT, Reliability Analysis, FMEA, on-line monitoring and off-line diagnosis tasks is essential. To this end, we are developing an integrated tool set, consisting of TEAMS, TEAMATE, TEAMS-RT and HARVESTER.

Third, we are exploring means of automating the model generation process. This includes generation and/or validation of models via simulation, extraction of information from legacy data formats and logistic database, and interfaces to CAD data bases.

Fourth, since most modern systems are increasingly software driven, modeling only the hardware for testability purpose is no longer adequate. We are developing techniques for functional modeling of complex systems involving hardware and software.

The use of the same model during DFT, Test Requirements Documentation (TRD), FMECA, TPS and PIMA development eliminates redundant modeling and ensures that the detection and isolation measures predicted in the design phase are realized in the field.

In the next section, we present a brief description of multisignal modeling, illustrating its benefits and simplicity with the model of a 1553 dual controller, dual bus system. This is followed by an overview of the Integrated tool set, the ID process and three real-world applications at different stages of the ID lifecycle. Finally, we present an overview of current projects at QSI: enhancements to the toolset, simulation based model generation and software testing and monitoring.

It is our goal to provide the customer with the most comprehensive and cost-effective ID solution. This paper provides the reader with a snapshot of our current offerings, and a roadmap to the future.

2. MULTISIGNAL MODELING

Minimizing the life-cycle cost of a system requires a well-coordinated effort involving people of different expertise. In effect, the model is the means by which people document and convey their understanding of the system, as it relates to their fields of expertise. For example, to the design engineer, the model could be a block diagram with transfer functions, whereas to a maintenance engineer, it is the schematic of replaceable components that make up the system. The objective is to develop a modeling methodology that is simple and intuitive enough so that people of various disciplines can understand and relate to it, yet powerful enough to be used during the entire life-cycle of a system.

2.1 BASIC OBSERVATIONS INSPIRING MULTISIGNAL MODELING METHODOLOGY

The foundation of multisignal modeling is based on the following observations:

First, for diagnostic purposes, we only need to model how a fault (or cause) propagates to the various monitoring points. The objective is not design verification: we assume that the system normally works to specification. The failure of one or more components (causes) results in system malfunctions (effects) that are observable at various points (test points) in the system. For FMECA, the goal is to trace the effects of the failure and assess their impact on system performance. For DFT, the goal is to ensure that the system is sufficiently observable (and controllable) so that the cause of a malfunction can be easily identified. In field maintenance, the goal is to identify the cause of a malfunction in minimum time/cost. In all these cases, it is sufficient to model the system in its failure space. Thus, the system can be modeled in terms of first-order cause-effect dependencies, i.e., how a faulty node affects its immediate neighbors. Higher-order dependencies can be inferred from first-order dependencies.

Second, the failure space is not binary (i.e., simple pass/fail), as is assumed in structural and traditional dependency models. The function space is multidimensional. Consequently, the failure space, which is the complement of function space, is also multidimensional. For example, the function of a sine wave generator is to generate a sine wave of specified amplitude, phase and frequency. It is said to have failed if the output sine wave does not have the desired amplitude, phase or frequency.

Third, since the failure state can be arbitrary, it is unnecessary to model the exact quantitative relationships. In order to illustrate this assertion, consider a cascade of three amplifiers, having gains of 2, 3, and 4, with an overall gain of 24. If, due to a fault, the new gain is 12, the first stage, with a design gain of 2, should not necessarily be implicated; the gain of any of the stages may have been reduced by half due to a failure. Thus, when the same attribute is modified by multiple components, quantitative relationships convey little, if any, information. If the gain is off, the amplifiers will be likely suspects. So, it is only necessary to identify the important functional attributes (or the dimensions of the function space) and associate them with the appropriate components and tests. These attributes are the signals.

Fourth, there can be two distinct types of failures: *functional* failures and *general* failures. Consider a lossless (passive) filter consisting of an inductor and a capacitor. If a fault in the inductor or capacitor causes a deviation in the center frequency, it is considered a functional failure, i.e., a fault that affects the function it was supposed to perform. On the other hand, if the fault is a short-circuit that causes the output power to be zero (i.e., a lossless filter causes a power-loss!), this is a general failure, that is, a catastrophic failure affecting attributes beyond its normal functioning by interrupting the flow of information through it. Thus, a failure in a module can either affect the attributes it was supposed to (functionally) modify, or all the attributes flowing through it. This affects how overall cause-effect dependencies are derived from the structure and signal information.

2.2 MULTISIGNAL MODELING VERSUS DEPENDENCY MODELING

Dependency modeling is a refinement on the structural modeling approach, where failure modes are added in an effort to model functional failures. In the filter example, the failure modes could be “out of tolerance'', causing a functional failure, and “short circuit'', causing a general failure. Thus, dependencies involving multiple signals are modeled as multiple single dependencies, one for each signal. This is done by breaking components into failure modes and considering their effects separately. The structural distortion in dependency models stems from the mapping of a multi-dimensional attribute (or “signal”) space of a physical system into a single-dimensional (dependency) space. Since this deviation stems from the judgement of the modeler based on some local reasoning, such dependency models are subjective with limited validity. This limitation, and the resultant validation problem, has rendered dependency modeling into an art. Test program developers, who inherit these

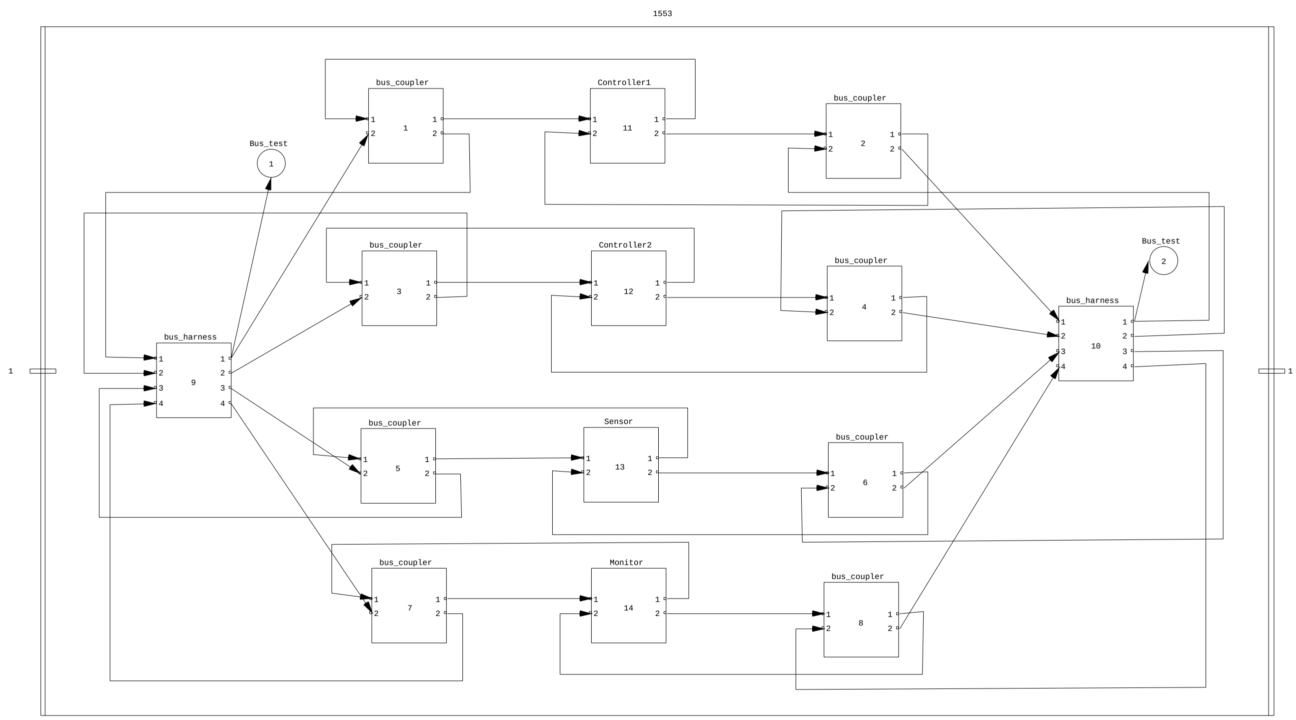

Figure 1: LRU level representation of the 1553 Bus (dual bus and controller).

models from the modeler, are unable to validate these models, and are consequently unwilling and/or unable to use the results in developing test programs.

Conceptually, a multi-signal dependency model is akin to overlaying a set of (single-signal) dependency models on the structural model, and, hence, the model corresponds closely to the schematics of the system. Thus, dependency modeling is a subset of multisignal modeling.

2.3 KEY ADVANTAGES OF MULTISIGNAL MODELING

Multisignal modeling captures the necessary useful and important knowledge about the system for fault diagnosis without being weighed down by unnecessary details (which drive up the model generation cost, as in exact simulation and/or qualitative reasoning) and computationally expensive simulations and/or reasoning techniques (which makes them impractical for large-scale systems applications). Furthermore, this modeling approach does not require the explicit knowledge of failure modes in a system. This means that the modeling approach enables the detection and isolation of unanticipated failure modes. Moreover, failure modes and effects analysis can be performed by specifying the signal-failure mode association for each component, if necessary.

The simplicity of the model allows for easy validation and acceptance. This is especially important since the same model is to be used by different people at different stages of the lifecycle of a product. In addition, since multisignal models do not require detailed design information, even products at conceptual stage can be modeled, analyzed and improved for testability.

2.4 AN EXAMPLE MODEL OF A 1553 BUS SYSTEM

The bus system is known to be a tricky system to model. This is because even though there is a physical connectivity between every device in the bus to every other device, normal communication is usually point-to-point between the controller and a device. Thus, the functional dependencies are point-to-point for functional failures. However, a malfunctioning device may disrupt all communications in the bus, resulting in an all-to-all dependencies for certain catastrophic or general failures. Due to these properties, modeling a bus system is extremely difficult in the traditional dependency modeling context. For this reason, Boeing Helicopters (BH) contracted QSI to demonstrate the benefits of multisignal modeling on a 1553 bus system.

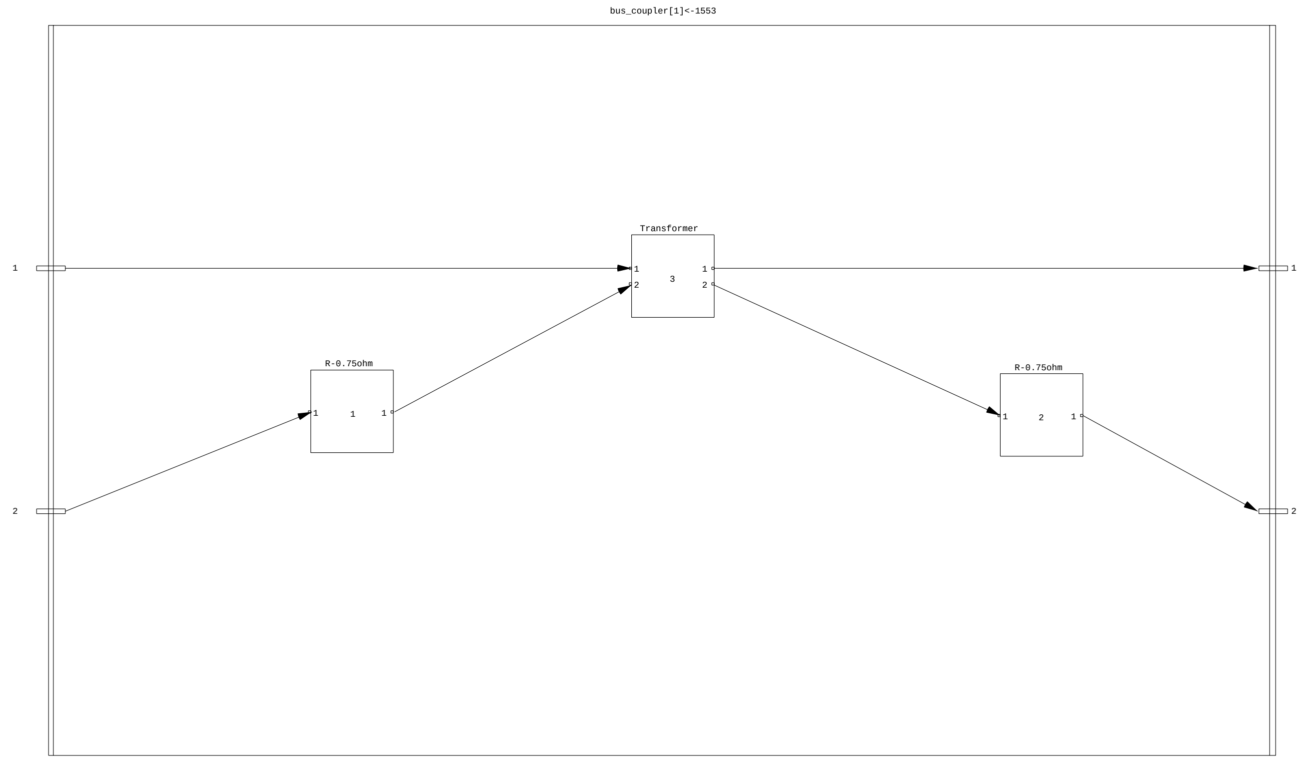

Figure 2: A simple 1553 bus coupler. Note the 0.75 series resistors to dampen inductance.

The key objective of this modeling effort was to illustrate that it is possible to stay close to structure and to model complex and dynamic dependencies. The close conformity to structure is also extremely useful in field maintenance, where LRUs, SRUs and components are replaced depending on the maintenance level. The same hierarchical model can be used in O-level, I-level, and D-level maintenance. With test levels, operations and resources assigned to tests, the diagnostic strategy can be dynamically tuned to different levels of instrumentation.

TEAMS also allows the modeling of fault-tolerant systems, without having to explicitly enumerate all relevant multiple failure combinations. Redundancies are modeled via AND nodes. The BH implementation of the 1553 bus system includes dual redundant buses and controllers. The system can operate in the dual bus mode or single bus mode (Bus A or Bus B), and can use either controller A or B. These mode switches also lend further insights into the health of the system, even during off-line maintenance. For example, in the Dual Bus mode, only 8% of the components can be isolated uniquely. The fault isolation number jumps to 95%, if the system is tested in Dual Bus, Bus A and Bus B modes in a controlled mode-sequence. Since a system should be exercised in all possible modes to maximize detection and isolation, a testability analysis tool that can model system configurations is of great value. The multisignal model of the 1553 bus presented here is a unified model for all possible operational modes of buses and controllers.

Due to space limitations, we will present only the salient features of the model in this paper. The full report [4] and the model is available for download from our web site at www.teamqsi.com .

Fig. 1 presents the LRU level representation of the 1553 bus. The entire model is made out of a handful of basic building blocks, such as the transmitter, the receiver and the bus couplers. The actual cabling for the bus is contained with in the modules called the *bus_harness*. The bus harnesses on either side and the corresponding *bus_coupler* show the Dual bus architecture. Four devices, Controller 1, Controller 2, Sensor and Monitor, are connected to both buses. Each of these devices will interface to the bus via dual channel receiver and transmitter and a pair of couplers (e.g., see Fig. 2 for Controller 1). As is evident from the symmetry of the model, it is extremely easy to add

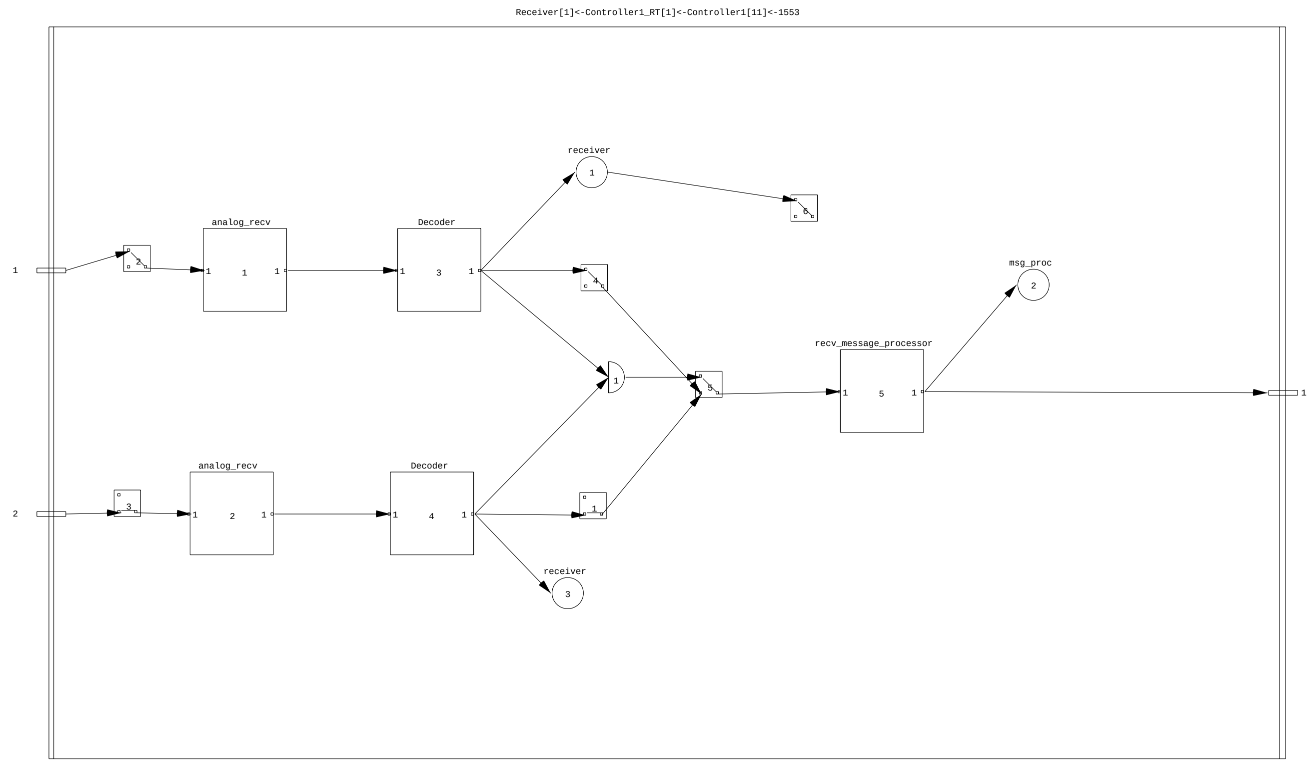

Figure 3: Screendump of the model of a receiver section of the RT in a 1553 dual bus setup.

new devices to the bus.

Our emphasis has been to model the basic characteristics, and illustrate that even though every device has a path, and consequently a potential dependency, to every other device, over 95% fault isolation can be achieved by modeling the characteristics of the devices via signals.

The signals considered for the overall model are the following: a) Attenuation, or poor gain; b) low Signal-to-noise-ratio (SNR) due to higher noise; c) AC and DC impedance; d) message integrity; and e) data integrity. For example, the important characteristics of the wires are: a) Attenuation per 100m, b) Noise pickup and shielding quality (SNR), and c) DC Impedance.

Fig. 2 presents a standard 1553 bus coupler, as defined in [5]. The primary side has series resistors to reduce the time constant. An ideal transformer is assumed. The transformer consists of the primary, secondary and the core, and is associated with the signals, AC impedance and Attenuation. The Core of the transformer blocks the DC signals (DC Impedance) and also provides electrical isolation (Blocks general failure). The resistors are associated with DC-Impedance signal.

The RT couplers are nearly identical to the bus coupler, except that they lack the 0.75 ohm series resistors. The same bus coupler and RT coupler is used for every device, promoting reusability of their models.

All devices interface to the bus using this transceiver unit, consisting of a Receiver (see Fig. 3), a Transmitter (not shown) and one RT-coupler for each bus. Even though all the RTs are considered identical, they each have a unique address, which distinguish one from the other. We assume that the Receiver is always on, so as to decode if the message is meant for itself or not. Hence, it is address-neutral. The RT couplers, being passive devices, are also address neutral. The transmitter, however, transmits only when it is its turn. The transmitted word also includes the address of the transceiver. Thus, the individual transmitters are distinguishable from each other, and hence are prefixed by the device name. Also, since they are functionally distinguishable from each other, the signals attached to them (message integrity, data integrity, SNR etc.) are also prefixed by the corresponding device names.

The Receiver (see Fig. 3) consists of the analog receiver (a low noise amplifier), a Decoder (that reconstitutes the digital information from analog input) and a message processor, that extracts the message words from the bit stream. Any error in reception can be recorded and transmitted back to the controller, as denoted by the test points.

When in dual bus mode, the message processor monitors both channels. If the word received from one channel has a parity error, it rejects that, and processes the word from the other channel. Thus, there has to be failure in both channels for the message processor to receive a corrupt word. This redundancy is modeled by the AND node.

However, when in single bus mode (i.e., Bus A or Bus B) fault tolerance is not possible. The three switches in the model then bypass the AND node and connect it directly to the appropriate channel. Thus, a single unified model is represents all three modes of operation.

The 1553 bus has software tests in the form of status words (e.g., parity error, word error etc.). We also add three special tests to check the integrity and quality of the bus. These tests require elaborate setup and operator intervention. Hence, they are assigned high costs (so that they are used as a last resort) and high test levels (to indicate that they may be available only in the Depot level maintenance facility). These tests are:

1. Continuity: Check for DC impedance and continuity
   Test level: 11
   Test Signals: DC_Impedance
   Operations: Attach multimeter in Bus
2. SNR: Check for Signal to noise ratio and noise pickup from wires.
   Test level: 12
   Test Signals: SNR
   Operations: Noise test setup [5].
3. Square_wave: Apply a square wave and check for rise time, fall time, oscillations, etc.
   Test level: 11
   Test Signals: AC_Impedance
   Operations: Measure AC impedance[5].

Testability Analysis of the model using TEAMS yielded the following results. Even though there is a physical path from every component to every other component, effective fault isolation is achieved in spite of the simplicity of the model. For example, the faulty LRU can be isolated over 90% of the time at the flight-line (O-Level) using only the all BIT and status word tests only. The isolation number drops to less than 10%, if isolation of a faulty component is attempted.

However, in a depot-level maintenance context, where the user-intervention type bus tests are available, most of the faults can be isolated to a faulty component. However, the maintenance cost and system downtime will be high.

To test a system properly, it should be exercised in multiple modes of operation. This can improve the diagnostic resolution dramatically. The most remarkable finding of the testability analysis on the 1553 model was that by exercising the system in the three modes of operation (Dual Bus, Bus A only, and Bus B only modes), the same diagnostic resolution is obtainable in-flight and for flight-line maintenance as is achievable in Depot level maintenance using the expensive user-intervention type tests. The capability to model and analyze a system with multiple modes of operation in a unified framework is a unique capability of TEAMS.

3. QSI INTEGRATED TOOLSET

QSI's integrated tool set automates the DFT, Reliability Analysis, FMECA, on-line monitoring and off-line diagnosis tasks (see Fig. 4). The software tool set consists of :

- TEAMS 4.0: Testability assessment and improvement (DFT), Reliability analysis, FMECA and pre-computed diagnostic test strategy generation in a variety of forms (e.g., SGML-based Interactive Electronic Technical Manual);
- TEAMS-RT: on-board diagnostics, health and usage monitoring systems;
- TEAMATE: Portable Intelligent Maintenance Aids (PIMAs) with interactive electronic technical manuals and multi-media animation, Dynamic TPSs for ATEs.
- HARVESTER: Scheduled and unscheduled maintenance and diagnostics data collection, statistical data analysis and data mining for trend and anomaly detection/isolation.

Each of these functional modules are briefly described in the following subsections:

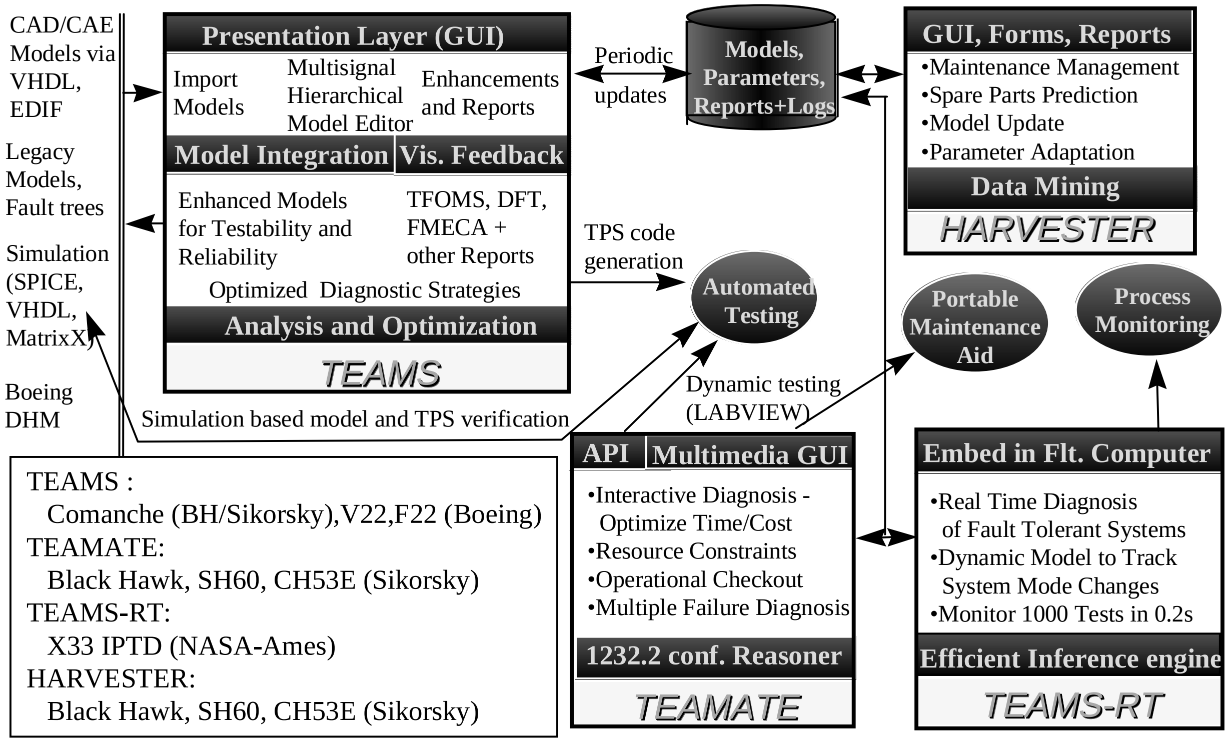

Figure 4: TEAMS, TEAMATE , TEAMS-RT and HARVESTER: The Integrated Tool set.

3.1 TEAMS

TEAMS, Testability Engineering And Maintenance System, is an X-windows-based software tool, that integrates the multi-signal flow graph modeling methodology and the algorithms in an easy-to-use graphical user interface [1]. TEAMS has been used for testability analysis of large systems containing as many as 50,000 faults and 45,000 test points. TEAMS minimizes the life-cycle cost of a system by aiding the system designer and test engineer in embedding testability features, including "built-in-test" requirements, into a system design; and by aiding the maintenance engineer by developing near-optimal diagnostic strategies (see Fig. 5).

TEAMS is used to: (i) model individual subsystems and integrate them into system models, (ii) analyze and quantify testability of systems and subsystems, visually pinpoint the diagnostic inefficiencies of a system, and make recommendations towards the design of completely testable systems, (iii) provide a comprehensive aid to automate the generation of FMECA reports, (iv) generate near-optimal diagnostic procedures for a variety of realistic testing options (see Fig. 6). Thus, TEAMS is mainly a DFT tool, but its (pre-computed, and, hence, static) diagnostic procedures can be embedded into Interactive Electronic Technical Manuals (IETMs) and Automatic Test Equipment (ATE).

Examples of problems that TEAMS can solve are:

- With a given set of tests, can all failures be detected; can they be isolated down to specific

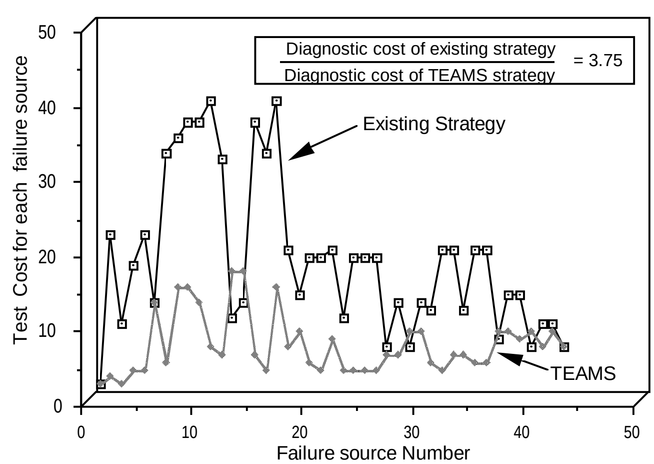

Figure 5: Comparison of test costs of existing and TEAMS generated strategies for the anti-collision system of Blackhawk helicopter

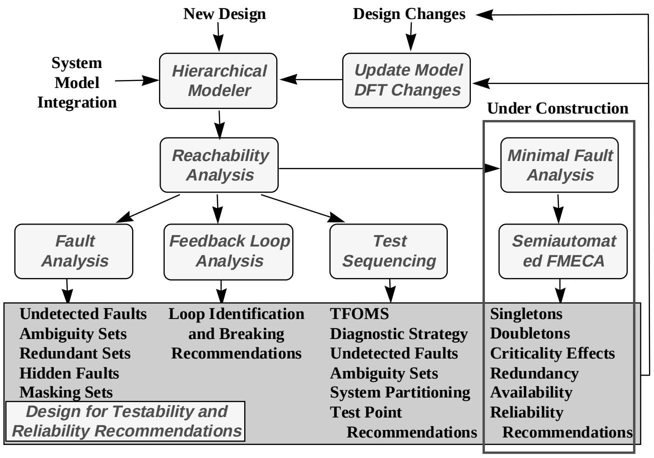

Figure 6: Analysis capabilities of TEAMS.

replaceable units, or only to a group of units?

- What tests should be used, and where should they be located, so that (on average) all faults can be isolated in minimal time and/or cost?
- What is the most efficient sequence of tests (and swap-outs) that will isolate the failure? (Important for TPS and field-test procedures.)
- What percentage of modules pulled as "faulty" are actually OK? (this has major impact on the logistics support system.)
- How can one (re)allocate system functions to replaceable units to get better testability properties of the full hardware/software design?
- Are all the components in the system reliable enough to survive the entire mission? Which components are most critical?

3.2 TEAMS-RT

TEAMS-RT (see Fig. 7) is a real-time companion tool to TEAMS for on-board diagnosis and on-line system health monitoring. It takes as inputs a TEAMS model of the vehicle and on-board smart-sensor processing results on system health (the results may be asynchronous). TEAMS-RT then identifies the known bad, known good and suspect set of components. Some unique features of TEAMS-RT are : (i) efficient real-time processing of sensor results, (ii) update of fault-test point dependencies in response to system mode changes, and (iii) update dependencies resulting from failures in redundant components. TEAMS-RT takes only 0.1s to process 1000 sensor results and diagnose multiple faults in a 1000 component system with 80 modes of operation. A distributed TEAMS-RT architecture may be embedded in flight computers to continuously monitor the system health, and identify in-flight failures. Examples of problems that TEAMS-RT can help solve are:

- What are the minimal number of sensors to be monitored to ascertain the system health?
- How do I monitor, diagnose and reconfigure a multi-mode system in real-time?

Table 1 presents simulation results for TEAMS-RT on a 1000x1000 system. Column 1 lists the number of faults inserted. |Tp| is the number of tests that passed

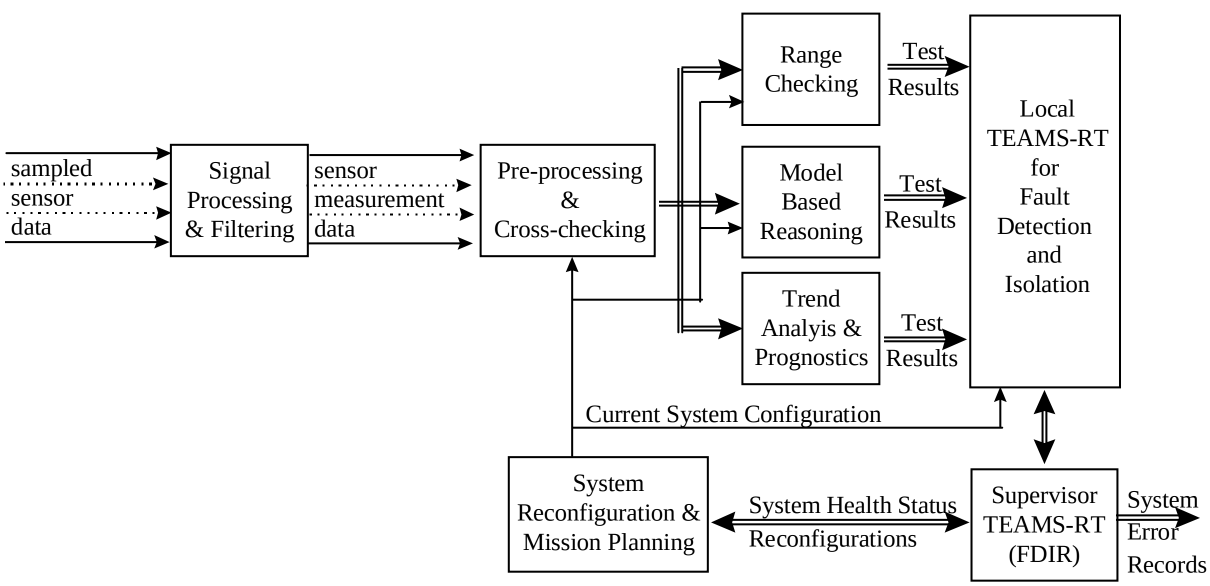

Figure 7:Real time process monitoring using TEAMS-RT.

#faults	|Tp|	Good	Bad	Suspected	Time(ms)
1	**993**	**997**	**1**	**2**	**50**
2	**978**	**996**	**2**	**2**	**50**
5	**931**	**991**	**5**	**4**	**50**
10	**881**	**983**	**10**	**7**	**75**
20	**819**	**973**	**20**	**7**	**87**

Table 1: Performance results of TEAMS-RT for simulated system with 1000 failure sources and tests.

in spite of the failures. The remaining columns list the number of components that were declared to be good, bad and suspected (residual ambiguity) by TEAMS-RT, and the processing time. Similar timings were observed in the X33-IPTD test-stand (see Section 4.3).

Key benefits of deploying TEAMS-RT in a Real-Time FDIR system are:

- Failures are detected/isolated quickly reducing risk of disasters.
- Even multiple failure combinations can be isolated quickly.
- Reduce “cannot duplicates” by detecting/isolating failures when they happen.
- The above information can be used to reconfigure systems and re-plan missions.
- Efficient condition based maintenance minimizes system downtime.

3.3 TEAMATE

TEAMATE, read team-mate, is a companion tool to TEAMS and TEAMS-RT for adaptive field diagnosis and dynamic TPS execution for automated testing (see Fig. 8). Thus, TEAMATE is an interactive diagnostic engine that takes the guesswork out of trouble shooting

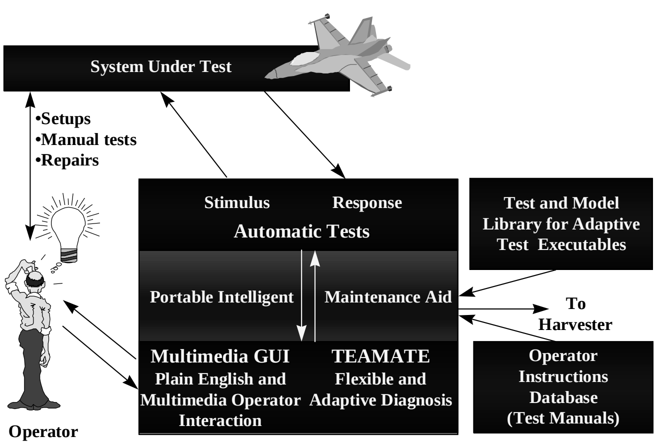

Figure 8: TEAMATE Deployment

by identifying the failure source(s) in the shortest possible time, subject to various constraints on available resources, setup operations already performed, the initial suspect set generated by TEAMS-RT and pilot squawks. The diagnostic engine can be integrated with a Portable Intelligent Maintenance Aid (PIMA) to assist field personnel in pre-flight checkouts and post-flight repairs. TEAMATE employs TEAMS generated export files to extract the system model to perform interactive diagnosis. Examples of some of the problems that TEAMATE can help solve are:

- How do I troubleshoot the system given different levels of instrumentation and resource constraints at different facilities?
- How can I use symptoms and results from TEAMS-RT to fine-tune my off-line diagnosis?

In real-life system operation, multiple failures can, and do, happen. In the following we compare two approaches [2] to troubleshooting in the presence of possibly multiple failures. The multi-fault capability of TEAMATE can significantly reduce false removals at the expense of some additional testing, as illustrated in Table 2.

	Single Fault Strategy			Multiple Fault Strategy		
Avg. # of failures	Retest OK	Test Cost	Num. Tests	Retest OK	Test Cost	Num. Tests
0.9	54.2%	14.9	16.1	0.5%	28.2	16.7
1.5	41.3%	15.0	17.7	0.1%	41.6	18.7

Table 2: Reduction of false removals (RTOKs) caused by multiple failures.

3.4 HARVESTER

The on-line diagnostic and maintenance information collected by TEAMATE (components repaired, repair times/costs, test costs/times, etc.) is archived by HARVESTER, a maintenance database tool that can be deployed at various maintenance sites. It also archives diagnostic/maintenance data from external maintenance management databases, legacy and provisioning databases, or through HARVESTER’s rich, user-friendly graphical interface forms. HARVESTER provides the important link to the promise of integrated diagnostic process. HARVESTER has various parameter estimation algorithms built into it for the analysis of maintenance data. It can be used to further refine a system model residing in the knowledge base of TEAMS with updated repair costs, repair times, component failure rates and diagnostic costs and times. In addition, the analysis algorithms of HARVESTER can be invaluable at the maintenance site, since they also provide predictions of optimal shelf-stocks of spares, and parts requirements. The reports generated

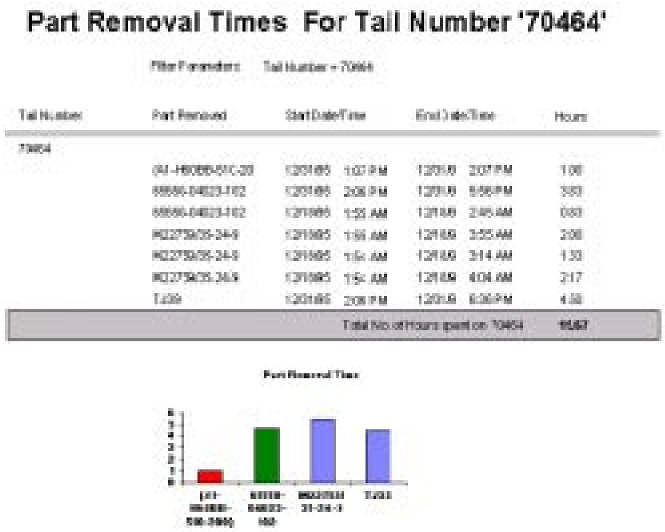

Figure 9: Sample screen dump from HARVESTER

by HARVESTER can provide insight into the bottleneck operations that slow down the entire maintenance process, by flagging unreasonably long task times, repeated failures of certain components, and updating hazard rates of components caused by changing environmental conditions. Examples of problems that HARVESTER can help solve are:

- What are the optimal shelf-stocks of spares?
- What are the bottleneck operations that slow down the entire maintenance process?
- How do I fine-tune my diagnostic strategies based on field failure data?

4. THE INTEGRATED DIAGNOSTICS PROCESS

4.1 THE ID PROCESS

A comprehensive solution should address all the facets of the problem. It should answers the following questions:

- Is the system designed to be maintainable ?
- If not, how can we improve it ?
- Can we prevent failures by detecting degradation?
- How can we continuously monitor it, and react quickly to failures, to prevent mishaps ?
- How can we repair it quickly to minimize downtime?
- Can we learn from experience and improve ?

The integrated diagnostics process using QSI's tool set consists of nine key steps:

1. The test engineer will import or build a multisignal model of the Unit Under Test (UUT) that captures the inter-connectivity among various components and their basic characteristics. The information required for multisignal models can be derived from a variety of sources. For example, the structure and inter-connectivity can be derived from CAD data - such as EDIF or structural VHDL, while behavioral information can be extracted from simulation data (PSpice and Hspice for analog circuits, behavioral VHDL, LABVIEW, $MATRIX_X$, MATLAB, and commercial simulators including Mentor Graphics, Cadence, Teradyne's LASAR, LogicVision & Viewlogic.).
2. The ID benefits resulting from QSI's Integrated Toolset is only as good as the underlying model. Thus, the model must be validated extensively through simulation, or reviewed by test engineers. In a simulation-based approach, faults are inserted and the "failed" tests for a given fault identified and compared with those predicted by the model. Any modeling errors will be corrected by the user, and the process is repeated until an accurate model is achieved.
3. Improve the testability of the system by identifying testability shortcomings and by trading off testability improvements against testability costs. TEAMS will aid the designer in this process by identifying DFT deficiencies and redundant tests, by assessing the percentage of faults that can be detected, and by identifying the ambiguity groups. If adequate fault isolation is not achieved, TEAMS will suggest new sensor placements and new test actions. Repeat steps 2 and 3 until the fault detection and isolation objectives are achieved.
4. Evaluate and improve system mission reliability via the reliability module of TEAMS.
5. Perform Failure Modes, Effects and Criticality Analysis (FMECA) by combining the system function (signal), redundancy and reliability data via TEAMS.
6. Use TEAMS-RT for real-time process monitoring using the same models. TEAMS-RT will process pass/fail outcome of tests in real-time to identify components that are good, that are failing, or have failed.
7. Use TEAMATE for field and depot level diagnosis. TEAMATE can also be used as a supervisor TEAMS-RT, to interact with the command module, and perform automatic (but intrusive ot off-line) tests. TEAMATE can be integrated with Technical documentation and embeded in PIMA, resulting in a smart diagnosis and training tool.
8. Collect field data via HARVESTER. The data can be analyzed to update model parameters such as failure rate of components, test costs, test times,

accuracy, test thresholds, etc. In addition, significant variables and trends can be identified using data mining techniques. Diagnostic error logs can also be used to identify modeling errors.

9. Use the refined models, resulting from feedback maturation in step 8, to improve diagnostic accuracy and efficiency in steps 3 -7.

4.2 IMPROVING LEGACY SYSTEMS USING QSI ID TOOLSET: OUR EXPERIENCE AT SIKORSKY AIRCRAFT

The legacy data (in SGML form), imported into QSI integrated tool set, has been successfully applied on Black Hawk, Sea Hawk and CH-53 helicopters. The diagnostic strategy from TEAMS, developed for a complex Blade-fold system on the Sea Hawk helicopter, was validated on a Navy training simulator in Jacksonville, FL in April 1996. Deployment of TEAMATE for Sea Hawk helicopters is expected during 1997-1998.

In the following, we illustrate the steps of the integrated diagnostic process, by way of application to an engine torque monitoring subsystem of the CH-53E helicopter.

1. The legacy data, in SGML format, for the engine torque monitoring subsystem was converted into intermediate file format (IFF). Using IFF → TEAMS interface module, a TEAMS model was created.
2. The test sequencing algorithms of TEAMS generated testability figures of merit (TFOMs) and an efficient off-line fault isolation strategy in SGML-format for the engine torque monitoring subsystem. The TFOMs were as follows: (i) percent fault detection and isolation are 100% each; (ii) mean-time-to-isolate is 8.02 units; (iii) there were four redundant tests (HT1, OT3-1, OT13 and OT14); (iv) test point efficiency is 64.26%; and (v) mean-time-to-failure of the subsystem is 5,707 hours. The SGML-diagnostic tree transparently interfaces with the local portable maintenance aid, consisting of *local* HARVESTER, TEAMATE and IETM.
3. The fault isolation strategy generated by TEAMS is only as good as the underlying model. In this simple case, the personnel in the technical services division of Sikorsky validated the model. For complex systems (e.g., the entire CH-53E), we suggest the use of simulation. Steps 1-3 were accomplished in less than ½ a day.
4. Since tests in a TEAMS model are categorized into different levels depending on whether they are executed on-board or on the ground, priority levels of tests and precedences, only those portions of the model corresponding to on-board monitoring can be *transparently* executed via TEAMS-RT. Based on our experience with X33 Integrated Propulsion

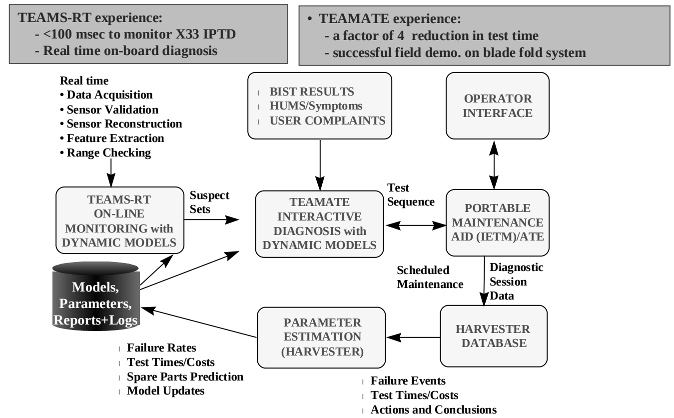

Figure 10: Role of TEAMS-RT, TEAMATE, and HARVESTER in lifecycle cost reduction.

Testbed Demonstrator Project, TEAMS-RT is ideally suited to filter out (typically voluminous) advisories and warnings, identify components requiring maintenance and isolate multiple failures.

5. The suspect set from HUMS and the flight log from the computer acted as triggers for ground-based fault diagnosis via HUMS ground station, consisting of a *site* HARVESTER (in our case, *site* HARVESTER and the *local* HARVESTER were the same). The HARVESTER database currently interfaces with the Aircraft Information System (AIS) database at Sikorsky Aircraft. Plans are underway to interface with NALCOMIS and other enterprise databases at Sikorksy as well.
6. The *local* HARVESTER, IETM and TEAMATE process the symptom and perform interactive diagnosis. TEAMATE is a very versatile and flexible model-based expert system for interactive fault diagnosis on the market today. For example, it allows *dynamic changes* in resource availability, test setup status, component status and test availability, block and sequential testing and multiple fault diagnosis. The diagnostic session data (failure events, test outcomes, repair actions, test times, repair times) are automatically collected by the *local* HARVESTER. The *local* HARVESTER transmits flight logs, parts removed and parts installed to AIS database.
7. The *central* HARVESTER collects data from site HARVESTERS to perform data mining. Preliminary data mining algorithms to update failure rates and for predicting spare parts requirements are developed. This is ongoing work.
8. Over time, the data collected by HARVESTER will be used to refine the models and improve diagnostic performance.

The above process conclusively demonstrated that faster, cheaper and better diagnostics can be generated using a model-based approach. The benefit is expected to be higher for larger, more complex systems, which are often beyond the realm of human troubleshooting strategies.

4.3 REAL TIME PROCESS MONITORING: OUR EXPERIENCE IN THE X33 IPTD

Onboard real-time fault detection, isolation and reconfiguration (FDIR) is essential for high performance vehicles and unmanned autonomous space vehicles. In the event of failures, the fault should be isolated as quickly as possible, and the system should be immediately reconfigured to contain any damaging effect of the failure. If the failure is in a redundant subsystem, the reconfigured system can continue with the mission, possibly in a degraded mode. The on-board FDIR must be responsive, decisive, and accurate.

Last year, NASA-ARC, NASA-LeRC, Rockwell Aerospace (now Boeing-NA) and QSI have applied TEAMS multisignal models and TEAMS-RT to the X33 Integrated Propulsion Test Demonstrator (IPTD). IPTD is a combination of pipes, valves and sensors that simulate the liquid hydrogen (LH_2) and liquid oxygen (LO_2) and helium (He) subsystems.

The objective of system health management is to monitor the system while in operation, extract meaningful features from the sensor data, detect and predict failures, and isolate any faults using TEAMS-RT.

The real-time prognostic and diagnostic capabilities of the integrated software were demonstrated on the IPTD during May 1996. In particular, the memory and real-time requirements of TEAMS-RT were significantly better than the specifications (memory : 0.9 MB versus 2 MB specified; time: 50-100 ms versus 200 ms specified). In addition, TEAMS-RT could update, in real-time, the fault-test dependencies in response to system mode changes (there were 44 modes in IPTD) and failures in redundant components.

This project conclusively proved that on-board real-time diagnosis of complex subsystems is feasible with a model-based approach. The superior memory and computational performance of the TEAMS-RT diagnostic engine on the IPTD suggests a distributed diagnostic architecture for the entire vehicle, and incorporation of fault recovery procedures in the face of failures.

4.4 MODEL-BASED TPS DEVELOPMENT: OUR EXPERIENCE IN THE JTIDS RECEIVER-SYNTHESIZER PROJECT

A model-based software environment for Test Program Set (TPS) development enables the generation of an optimized and reusable TPS, plus automatic validation of the TPS. System life-cycle cost is reduced via: a) a systematic approach to develop efficient TPSs, b) easy updates to TPSs spawned by design changes, c) improved ATE throughput by reduction of system test time, and d) reduced sparing costs by minimizing RTOKs.

AIL, Inc. (Deer park, NY), with assistance from QSI, has employed TEAMS 4.0 to generate, validate and demonstrate a test program set on a CASS station for the receiver-synthesizer board of the Joint Tactical Information Distribution System (JTIDS). The project was sponsored by the Space and Warfare Systems (SPAWAR) Command.

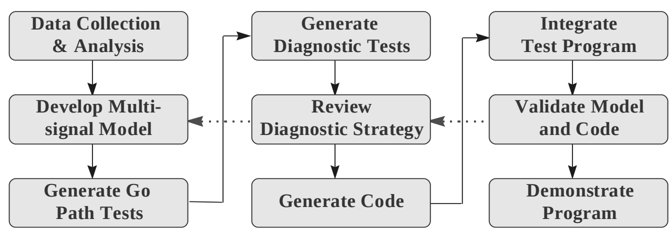

Figure 11: Model based TPS development in the JTIDS Receiver-Synthesizer project

The existing Test Program Set development process being utilized for the JTIDS avionics box and circuit boards is a manual, labor intensive effort prone to errors and subject to inordinate delays and cost growths as personnel changes occur. The TEAMS-based process (see Fig. 11) employed a multisignal model representation of the UUT. This representation of data would be used throughout the life-cycle of the Test Program so as to track changes as they occur to the UUT design, as well as the Test Program.

The results demonstrated that: (1) the number of tests needed was approximately 1/3 of the tests needed by the original manual test program (211 versus 568); (2) less ambiguity groups; (3) quicker validation using TEAMS diagnostic tree (13 faults per day were validated on the test station versus 7 for the original program; and (4) a potential 66% reduction in TPS development cost over manual methods is achievable.

5. ONGOING DEVELOPMENT ACTIVITIES AT QSI

Current development activity at QSI is focussed on three main areas: modeling, tele-diagnostics and software testing. In the following, we briefly outline our efforts in these areas.

5.1 MODELING DEVELOPMENTS

The benefits of the ID process are realized only if the underlying models are accurate. For example, a TPS will be error free, provided the underlying model is reliable. *Thus, the burden of validation shifts from validation of diagnostics, to validation of models.* Even though the multisignal models in TEAMS correspond closely to structure, and are easier to understand and verify compared to test sequences, a simulation based automated validation environment will greatly improve the confidence in the model. We are developing a software environment for automated statistical validation of TEAMS models via simulation.

The promise of lifecycle cost savings have not attracted many commercial users to ID process. Often, the upfront cost of dependency modeling offsets any potential future savings. Consequently, the Integrated Diagnostics is not viewed as a cost-effective option by many (e.g., the automotive industry). However, the modern automobiles are sensor rich, and have sufficient on-board processing power for real-time monitoring and diagnostics. They are also designed in CAD/CAE environments, Consequently, models of systems in Saber (by Analogy) and Matrix$_X$ (by ISI), are usually available. Current efforts at QSI are aimed at automatically extracting dependency models from Saber, Matrix$_X$, Pspice, etc. via simulation, and combining such models into system level models. This would greatly reduce the cost of modeling for ID, while improving its accuracy.

The cost of modeling can also be reduced by promoting reusability of models. We are developing a web/database based model management system to create and maintain test and module objects that can be shared among a large workgroup over a wide-area network. This will, for example, promote re-use of test procedures within and across test programs of UUTs. The goal is to provide an intuitive graphical user interface to the test engineer so that he can itemize the actual test and measurement relationships. This will include definition of the nature of input stimuli, any required prior setups, instrumentation, post setups, precedences and exclusions, functional grouping, as well as the actual measurement procedure. By clearly identifying all the steps, a test procedure can be treated as a standalone object. These objects can then be re-used and re-sequenced by the test sequence optimization algorithms of TEAMS.

5.2 NETWORKED DIAGNOSIS

There are two major motivations for networked diagnosis. First, large multi-national aerospace companies market, sell and service their products all over the world. They would like to leverage their organization-wide expertise to aid in diagnosis of aircraft deployed even in the most remote locations (see Fig. 12). And second, they would like an efficient information collection and distribution system, to make sure all the service personnel receive up-to-date information. Traditionally, technical documentation and service bulletins are issued to field maintenance personnel every six months. In a networked environment, this lag in information distribution may be practically eliminated.

We are currently making TEAMATE and TEAMS-RT network and database aware. In addition to the more traditional PIMA implementation, the resultant client-server architecture will also allow remote users to connect their web browsers to a central site, transparently download the GUI applet and multimedia technical and training documentation, and execute a tele-diagnostic session. This essentially transforms any web browser into a PIMA. Using our toolset, companies will be able to implement intelligent help desks that employ model based reasoning for in-warranty and pay-per-use services.

The role of HARVESTER is being expanded beyond that of a data collection and feedback analysis tool, to include active configuration management of software, models, logistics information and field data. For

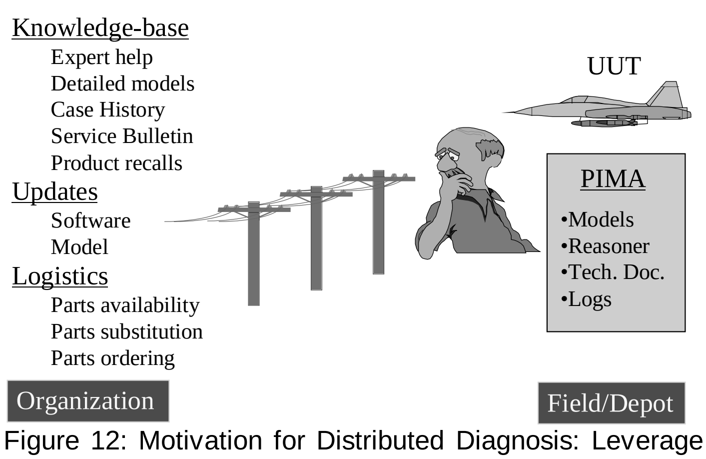

Figure 12: Motivation for Distributed Diagnosis: Leverage Organization-wide Diagnostic capability

example, HARVESTER can be used to actively *push* revised models, new features, bug fixes, service bulletins, safety advisories, and product recalls to remote sites, while pro-actively collecting (*pull*) failure events, maintenance logs, and anomaly reports.

5.3 Software Testing

Most complex modern systems are now a blend of hardware and software. Consequently, the testability analysis of a system is incomplete without adequately accounting for the effect of software.

However, software Testing is an extremely challenging problem. This is because:

1. Software is usually much more complicated than hardware.
2. Software flaws are design flaws. Consequently, two identical software programs will fail identically. This is in contrast to the *random* failure modes for hardware, where the possibility of two identical parts failing simultaneously is remote.
3. In the absence of “known good” response, design of tests is difficult. Usually, a test in a software involves some statistical or heuristic check or assertion on a variable, or computing an inverse relationship, or computing the same variable using alternative means (n-version programming).

Consequently, it is impossible to guarantee a software to be bug-free. We may only check it out over a large function space, and provide some statistical quality assurance.

We are applying functional testing concepts that are proven in testability and diagnosis of large complex systems, to solve the software testability problem. This involves four major steps:

1. Instrument the software so that bugs are detectable. This is similar to allocating test points in DFT.
2. Design stimuli so that bugs are “tickled” - i.e., crash, hang, produce garbage, or inaccurate results. This is equivalent to design of input patterns in hardware testing
3. Design tests to recognize wrong results. This typically involves comparing the measured value to a known good response
4. Based on test results, identify the faulty code segment. This step is identical to the fault isolation procedure employed in TEAMS, TEAMATE and TEAMS-RT

In combination with a structured approach involving tight specifications and automatic code generation, much of the difficulties in Software testing can be alleviated. We are currently applying our software testing techniques to the 12-ft Pressure Wind Tunnel at NASA Ames, where we are remotely monitoring real-time C-code generated by Matrix_X Autocode.

6. CONCLUSIONS

In this paper, we presented a comprehensive review of QSI’s ID tool set, reviewed some of our recent experiences in solving real-world problems, and outlined our current development efforts. We hope we have a comprehensive solution that will make Integrated Diagnostics commercially viable,

7. REFERENCES

1. Deb, S., Pattipati, K.R., Raghavan, V., Shakeri, M., and Shrestha, M., “Multi-Signal Flow Graphs: A novel Approach for System Testability Analysis and Fault Diagnosis,” in *Proc. IEEE AUTOTESTCON*, Anaheim, CA, pp. 361-373, Sept. 1 994.

2. Shakeri, M., Pattipati, K., Raghavan, V., Patterson-Hine, A., and Kell, T.,” Sequential Test Strategies for Multiple Fault Isolation”, *1995 IEEE AUTOTESTCON*, Atlanta, GA, Aug. 1995.

3. Sen, S., Nath, S.S., Malepati, V.N., and Pattipati, K., “Simulation-based Testability Analysis and Fault Diagnosis,” *1996 IEEE AUTOTESTCON*, Dayton, Ohio, September 1996.

4. Deb, S. “Diagnostic Modeling of a Data Bus System and Software using TEAMS 4.0”, final report submitted to Boeing Helicopters, PA.

5. “MIL-STD-1553 Designers guide”, Fourth Edition, ILC Data Device Corporation, Bohemia, NY.